

# UNIX - FILE SYSTEM BASICS

<http://www.tutorialspoint.com/unix/unix-file-system.htm>

Copyright © tutorialspoint.com

A file system is a logical collection of files on a partition or disk. A partition is a container for information and can span an entire hard drive if desired.

Your hard drive can have various partitions which usually contains only one file system, such as one file system housing the / file system or another containing the /home file system.

One file system per partition allows for the logical maintenance and management of differing file systems.

Everything in Unix is considered to be a file, including physical devices such as DVD-ROMs, USB devices, floppy drives, and so forth.

## Directory Structure:

Unix uses a hierarchical file system structure, much like an upside-down tree, with root (/) at the base of the file system and all other directories spreading from there.

A UNIX filesystem is a collection of files and directories that has the following properties:

- It has a root directory (/) that contains other files and directories.
- Each file or directory is uniquely identified by its name, the directory in which it resides, and a unique identifier, typically called an inode.
- By convention, the root directory has an inode number of 2 and the lost+found directory has an inode number of 3. Inode numbers 0 and 1 are not used. File inode numbers can be seen by specifying the -i option to ls command.
- It is self contained. There are no dependencies between one filesystem and any other.

The directories have specific purposes and generally hold the same types of information for easily locating files. Following are the directories that exist on the major versions of Unix:

Directory	Description
/	This is the root directory which should contain only the directories needed at the top level of the file structure.
/bin	This is where the executable files are located. They are available to all user.
/dev	These are device drivers.
/etc	Supervisor directory commands, configuration files, disk configuration files, valid user lists, groups, ethernet, hosts, where to send critical messages.
/lib	Contains shared library files and sometimes other kernel-related files.
/boot	Contains files for booting the system.
/home	Contains the home directory for users and other accounts.
/mnt	Used to mount other temporary file systems, such as cdrom and floppy for the CD-ROM drive and

	floppy diskette drive, respectively
/proc	Contains all processes marked as a file by process number or other information that is dynamic to the system.
/tmp	Holds temporary files used between system boots
/usr	Used for miscellaneous purposes, or can be used by many users. Includes administrative commands, shared files, library files, and others
/var	Typically contains variable-length files such as log and print files and any other type of file that may contain a variable amount of data
/sbin	Contains binary (executable) files, usually for system administration. For example <i>fdisk</i> and <i>ifconfig</i> utilities.
/kernel	Contains kernel files

## Navigating the File System:

Now that you understand the basics of the file system, you can begin navigating to the files you need. The following are commands you'll use to navigate the system:

Command	Description
cat filename	Displays a filename.
cd dirname	Moves you to the directory identified.
cp file1 file2	Copies one file/directory to specified location.
file filename	Identifies the file type (binary, text, etc).
find filename dir	Finds a file/directory.
head filename	Shows the beginning of a file.
less filename	Browses through a file from end or beginning.
ls dirname	Shows the contents of the directory specified.
mkdir dirname	Creates the specified directory.
more filename	Browses through a file from beginning to end.
mv file1 file2	Moves the location of or renames a file/directory.
pwd	Shows the current directory the user is in.
rm filename	Removes a file.
rmdir dirname	Removes a directory.
tail filename	Shows the end of a file.
touch filename	Creates a blank file or modifies an existing file's attributes.

whereis filename	Shows the location of a file.
which filename	Shows the location of a file if it is in your PATH.

You can use [Manpage Help](#) to check complete syntax for each command mentioned here.

## The df Command:

The first way to manage your partition space is with the df (disk free) command. The command df -k (disk free) displays the disk space usage in kilobytes, as shown below:

```
$df -k
Filesystem      1K-blocks      Used   Available Use% Mounted on
/dev/vzfs        10485760    7836644    2649116   75% /
/devices          0          0          0    0% /devices
$
```

Some of the directories, such as /devices, shows 0 in the kbytes, used, and avail columns as well as 0% for capacity. These are special (or virtual) file systems, and although they reside on the disk under /, by themselves they do not take up disk space.

The df -k output is generally the same on all Unix systems. Here's what it usually includes:

Column	Description
Filesystem	The physical file system name.
kbytes	Total kilobytes of space available on the storage medium.
used	Total kilobytes of space used (by files).
avail	Total kilobytes available for use.
capacity	Percentage of total space used by files.
Mounted on	What the file system is mounted on.

You can use the -h (human readable) option to display the output in a format that shows the size in easier-to-understand notation.

## The du Command:

The du (disk usage) command enables you to specify directories to show disk space usage on a particular directory.

This command is helpful if you want to determine how much space a particular directory is taking. Following command would display number of blocks consumed by each directory. A single block may take either 512 Bytes or 1 Kilo Byte depending on your system.

```
$du /etc
10    /etc/cron.d
126   /etc/default
6     /etc/dfs
...
$
```

The `-h` option makes the output easier to comprehend:

```
$du -h /etc
5k    /etc/cron.d
63k   /etc/default
3k    /etc/dfs
...
$
```

## Mounting the File System:

A file system must be mounted in order to be usable by the system. To see what is currently mounted (available for use) on your system, use this command:

```
$ mount
/dev/vzfs on / type reiserfs (rw,usrquota,grpquota)
proc on /proc type proc (rw,nodiratime)
devpts on /dev/pts type devpts (rw)
$
```

The `/mnt` directory, by Unix convention, is where temporary mounts (such as CD-ROM drives, remote network drives, and floppy drives) are located. If you need to mount a file system, you can use the `mount` command with the following syntax:

```
mount -t file_system_type device_to_mount directory_to_mount_to
```

For example, if you want to mount a CD-ROM to the directory `/mnt/cdrom`, for example, you can type:

```
$ mount -t iso9660 /dev/cdrom /mnt/cdrom
```

This assumes that your CD-ROM device is called `/dev/cdrom` and that you want to mount it to `/mnt/cdrom`. Refer to the `mount` man page for more specific information or type `mount -h` at the command line for help information.

After mounting, you can use the `cd` command to navigate the newly available file system through the mountpoint you just made.

## Unmounting the File System:

To unmount (remove) the file system from your system, use the **umount** command by identifying the mountpoint or device

For example, to unmount `cdrom`, use the following command:

```
$ umount /dev/cdrom
```

The `mount` command enables you to access your file systems, but on most modern Unix systems, the automount function makes this process invisible to the user and requires no intervention.

## User and Group Quotas:

User and group quotas provide the mechanisms by which the amount of space used by a single user or all users within a specific group can be limited to a value defined by the administrator.

Quotas operate around two limits that allow the user to take some action if the amount of space or number of disk blocks start to exceed the administrator defined limits:

- **Soft Limit:** If the user exceeds the limit defined, there is a grace period that allows the user to free up some space.

- **Hard Limit:** When the hard limit is reached, regardless of the grace period, no further files or blocks can be allocated.

There are a number of commands to administer quotas:

Command	Description
quota	Displays disk usage and limits for a user or group.
edquota	This is a quota editor. Users or Groups quota can be edited using this command.
quotacheck	Scan a filesystem for disk usage, create, check and repair quota files
setquota	This is also a command line quota editor.
quotaon	This announces to the system that disk quotas should be enabled on one or more filesystems.
quotaoff	This announces to the system that disk quotas should be disabled off one or more filesystems.
repquota	This prints a summary of the disc usage and quotas for the specified file systems

You can use [Manpage Help](#) to check complete syntax for each command mentioned here.